

Job entry subsystems have been developed to provide operating systems with an interface for managing some of the workload of computing facilities. One of the job entry subsystems for OS/VS2 has been further enhanced with the addition of a network job entry facility that allows full access to a network of computers in a manner consistent with a local operation. This paper discusses the design objectives, implementation, and extensions of that facility.

Network job entry facility for JES2

by R. O. Simpson and G. H. Phillips

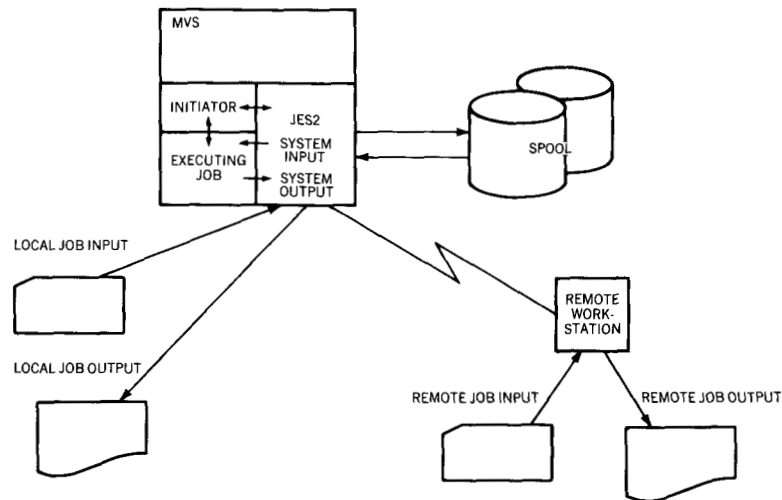
As computing systems evolved to incorporate virtual storage, operating systems and their subsystems were extended to make use of this capability.¹ The development of networking in data processing has allowed additional facilities to be available for use with operating systems.

With the advent of OS/VS2 Release 2, the Houston Automatic Spooling Priority system (HASP),² which was essentially an add-on package for earlier operating systems, became an integral part of the operating system. Known as Job Entry Subsystem 2 (JES2), it is an interface between the users of the computing system and the programs that make up the operating system.

JES2 is responsible for the input of batch computing work ("jobs") through locally attached devices such as card readers, through devices attached remotely by way of teleprocessing lines, and through a program interface known as the internal reader. JES2 queues batch work and selects work for execution. It manages executing jobs, system tasks such as VTAM (Virtual Telecommunications Access Method) or IMS (Information Management System), users logged on to TSO (Time Sharing Option), and their respective unit record (card image, print line, etc.) input and output (system input and system output) files. It distributes output files to local or remote devices that provide hard copy, such as printers, or to other media.

Copyright 1978 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

Figure 1 Typical MVS/JES2 configuration



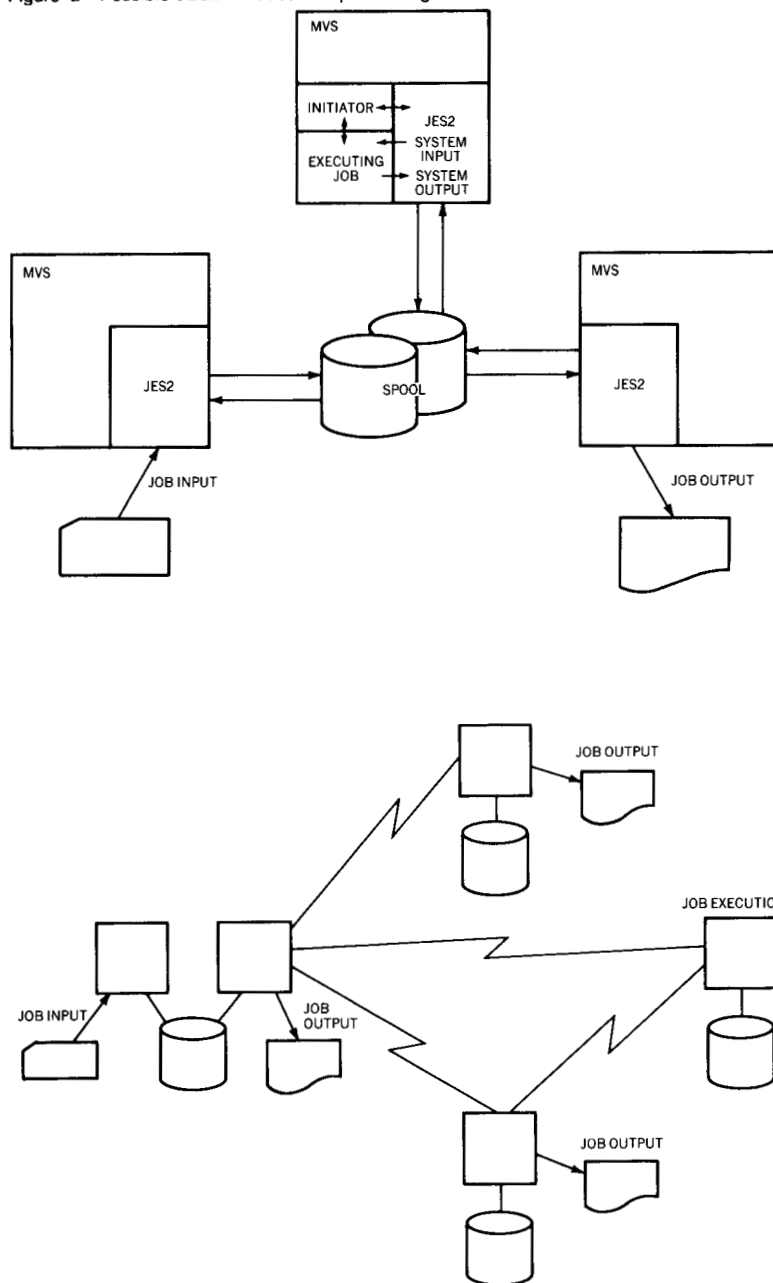
JES2 maintains its input and output data sets and work queues on high-performance direct access devices known collectively as the JES2 spool (from "simultaneous peripheral operation on line"). Through spooling, JES2 (a) shares essentially nonsharable unit-record-type input and output devices, (b) matches speeds between slower input/output (I/O) devices and programs executing in high-performance central processors, and (c) queues many types of batch work to allow for effective use of central processors by executing a broad mix of work simultaneously.

To the OS/VS2 user who submits a job for execution, JES2 is a major interface with the operating system (Figure 1). JES2 reads the job, controls its selection for execution, supplies input data to the programs invoked, collects the output data, and prints the output. Through job control language statements and JES2 control statements, the user tells JES2 how to queue the job for execution and where to send the output.

Through the JES2 Multiaccess Spool Facility (Figure 2), a user may have JES2 run a job on any one of several loosely coupled processors sharing a common job queue and spool, even though the input device through which the job is submitted is attached to only one processor. Likewise, JES2 may be directed to print or punch output on devices that are connected to any of the processors, not necessarily to the processor that executed the job. A JES2 Multiaccess Spool Facility user does not need to be aware of the way in which the processors and I/O devices are connected.

The addition of the Network Job Entry Facility (NJE) to JES2 makes a large amount of computing capability accessible to users at geographically dispersed locations (Figure 3). Each individual

Figure 2 Possible JES2 multiaccess spool configuration



JES2 system and each JES2 Multiaccess Spool Facility becomes a node in the NJE network; users at each node have access to the computing facilities of all the other nodes. Releases 1 and 2 of NJE for JES2 support interconnection of the nodes by way of binary synchronous communication (BSC) lines and channel-to-channel adapters. The BSC lines and channel-to-channel adapters are

driven directly by JES2; no additional telecommunications access method is required. Release 3 adds support for Systems Network Architecture³ (SNA) communications between NJE nodes using the Advanced Communications Functions⁴ (ACF) of VTAM⁵ (ACF/VTAM) and the Network Control Program⁶ (ACF/NCP).

Design objectives of the Network Job Entry Facility for JES2

The overall goal of NJE for JES2 is to provide users with access to the computing facilities of an entire network in a manner that is consistent with the local facilities provided by OS/VS2 and JES2. The following are some points that were considered during the design and implementation of NJE.

simplicity for end user	One of the design goals was to maintain the transparency of the computing configuration to the user. The user should view the job network as an extension of a local computing facility and not be concerned with the manner in which the various processors and I/O devices are connected. This goal was achieved by NJE; in fact, a user can submit a job through a local or remote card reader and later collect the output of the job at a local or remote printer without being aware that the job was actually transmitted somewhere else for execution.
ease of installation	In addition to being easy to use from the standpoint of a job submitter, it was felt that NJE should be no more difficult to install from a system programming standpoint than JES2 Remote Job Entry (RJE). That is, no elaborate predefinition of the network configuration should be required. When an NJE system is connected to an active NJE network, it should be able to determine the current network configuration and take appropriate action without requiring operator intervention or a static network configuration. This goal was met through the implementation of the network path manager, discussed later.
dynamic network configuration	NJE was designed to take full advantage of the communications facilities available to a job network without requiring preplanning of alternate routes to be used in case of line or central processor (node) failures. Again, this design goal was achieved by the network path manager, which becomes aware of disconnections (intentional or accidental) and connections of lines between nodes as they occur and uses this information to adjust the routing algorithms of NJE dynamically. Operator intervention is not required to reroute traffic around nodes or lines that are down. Likewise, entirely new lines such as temporary dial-up connections are automatically incorporated into the active job network.

A job network that can consist only of systems running JES2 is of limited utility, both for migration from one operating system to another and for long-term operation. Thus, a common set of line protocols and data formats was decided upon in designing job networking packages for JES2, HASP, ASP, and VM/370. The MULTI-LEAVING protocol⁷ originally developed for HASP to support remote job entry terminals using BSC lines was already common to the four systems. MULTI-LEAVING allows multiple independent data streams to flow in both directions at once over a single BSC line. An extension to this protocol was designed for job networking. A common set of control blocks, independent of any specific job entry subsystem, was invented to describe transmitted jobs and I/O files or data sets. Provision was made for a common form of network account number, to be associated with each job, which can be processed by any of the different job networking systems.

**interconnect
ability**

Because the dynamic routing mechanism (network path manager) presently exists only in NJE, a means for prespecifying the configuration of a network containing non-JES2 systems was designed into NJE. In this manner, a network consisting of all four of the different job networking packages (such as IBM's internal development laboratory network depicted in Figure 4) can be connected.⁸ The networking packages for HASP, ASP, and VM/370 that can communicate with NJE and with each other are known collectively as the Network Job Interface, or NJI.⁹

The common set of control blocks through which NJE communicates with other nodes in a job network was designed to allow extension by users for added functions and by IBM in future releases of NJE. Adding information to a control block does not require that similar changes be made simultaneously through the network. NJE is insensitive to the presence or absence of user-dependent information and does not depend on a fixed length for any of the common control blocks.

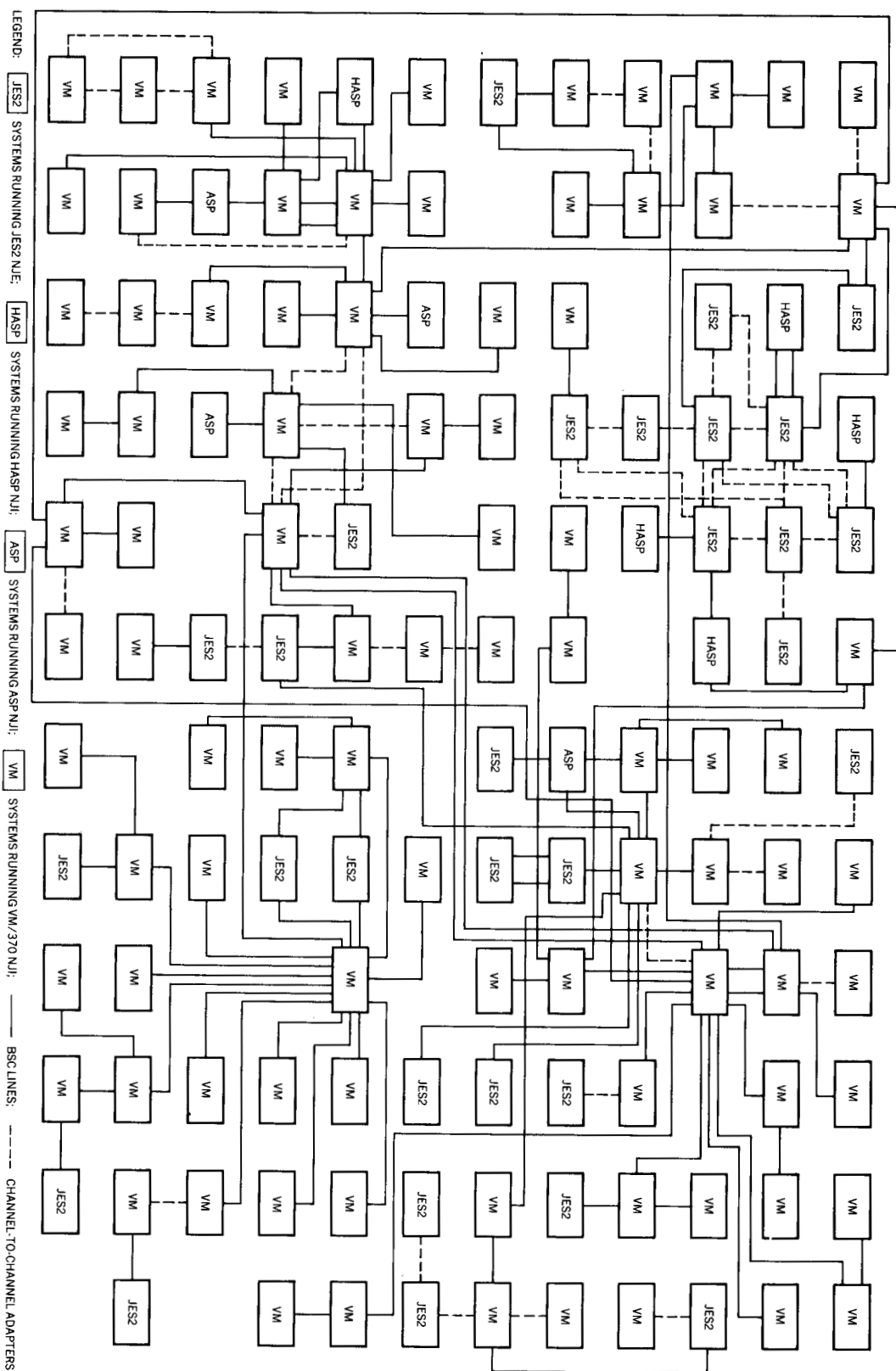
**extensibility
and
compatibility**

Because independence from the control block structure and size was achieved, it is possible for any given release of NJE to communicate with any other release using transmission facilities that are common to both. Thus, support for binary-synchronous NJE is retained in the release of NJE that supports SNA networking, to provide a one-step-at-a-time migration path and to allow communication with older systems such as HASP that do not support SNA.

NJE for JES2 was designed to function so that jobs and output data are not lost in the event of the loss of a transmission link or network node during transmission. This goal was achieved by a technique classically used by store-and-forward systems: during each store-and-forward operation for a job or output file, the trans-

recovery

Figure 4 IBM development laboratory network



mitter remains responsible for the entire job or file until it receives positive acknowledgment to the end-of-file message from the receiving node. If an error occurs during transmission, the receiving node purges its partially received job or output and the transmitting node requeues the work for retransmission from the beginning. For the first releases of NJE, there is no provision for restarting an interrupted transmission at the point of interruption.

Network job entry implementation

A job submitted to JES2 consists of job control language (JCL) statements and possibly input data sets. The input data sets, if present, are called in-stream data sets because they are included as part of the input stream to JES2. In-stream data sets are interspersed among JCL statements for a job. As the input service routine of JES2 reads a job, it places the JCL in a single file and each of the in-stream data sets in a separate file on the JES2 spool. The JCL file will be read by the OS/VS2 JCL converter in preparation for the job's execution. During the job's execution, JES2 will present the in-stream data sets as input to the programs that the job invokes.

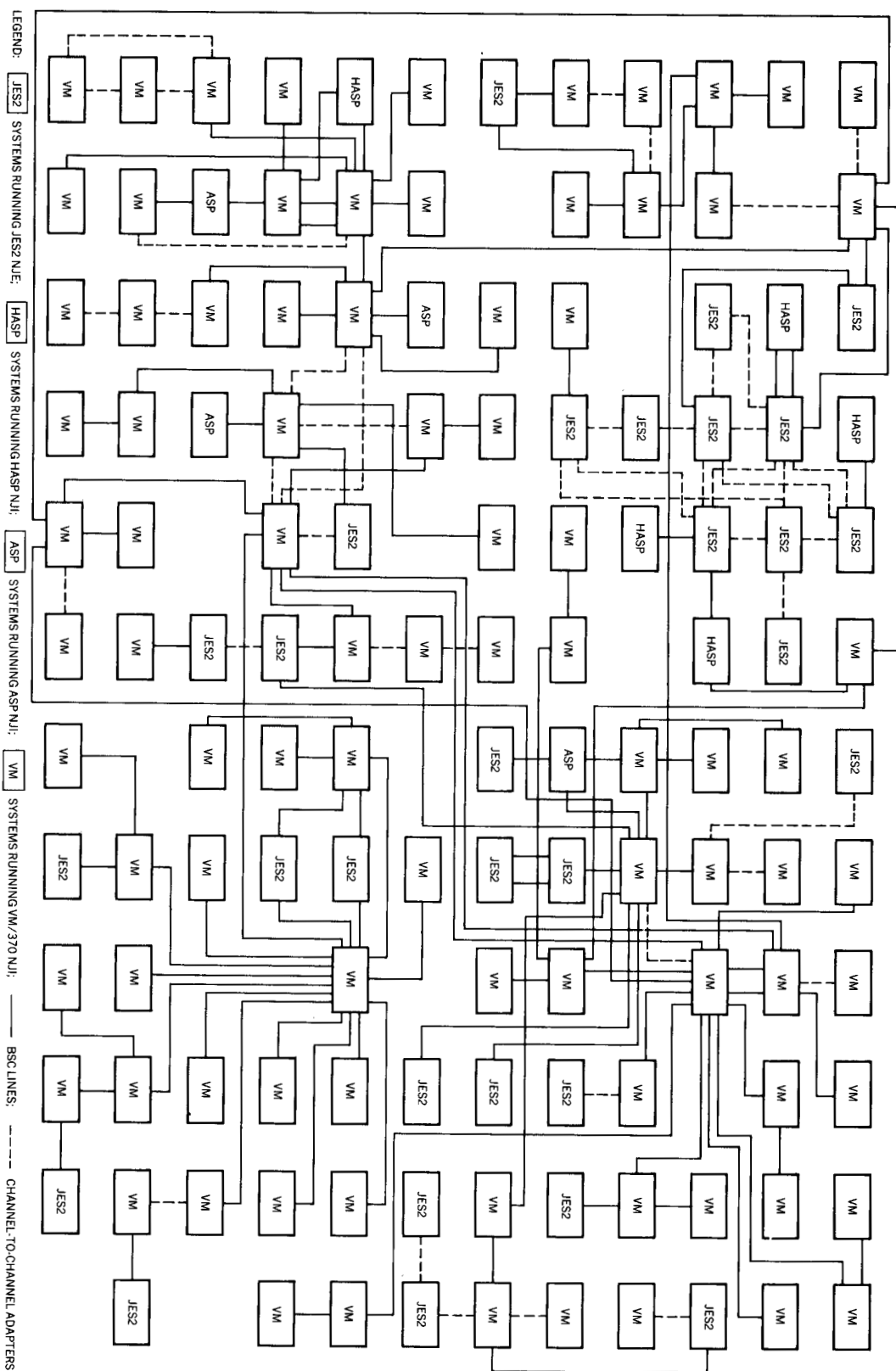
As a job executes, it can produce one or more system output data sets. These data sets are referred to as system output because the job entry subsystem (JES2) is responsible for their eventual disposition—to printers, punches, or other local or remote output devices, or to interactive terminal users. Some system output data sets, such as the JES2 job log, are produced automatically by OS/VS2 and JES2 without involving the programs invoked by the job. Most system output data sets are created by the executing programs as a record of their processing; most commonly they are listing files of some type.

When a job has been placed on the JES2 spool but is not actually in execution, it can be thought of as a collection of files: a JCL file and (perhaps) some input files before execution, a set of output files after execution. To perform a job networking function, JES2 must be able to send and receive these collections of files. To provide these capabilities, the following components have been added to JES2: (a) network path manager, (b) job transmitter, (c) job receiver, (d) system output transmitter, (e) system output receiver, and (f) command and message forwarder.

The network path manager is present in each copy of NJE for JES2. Collectively the path managers provide information to the components of JES2 on best paths and acceptable alternate paths on a dynamic basis. For JES2 components such as the command processor or console services, the information indicates the best NJE connection that should be used to reach the desired NJE destination node. In this case, the connection could be over a communications line, a channel-to-channel adapter, or across the spool

**network
path
manager**

Figure 4 IBM development laboratory network



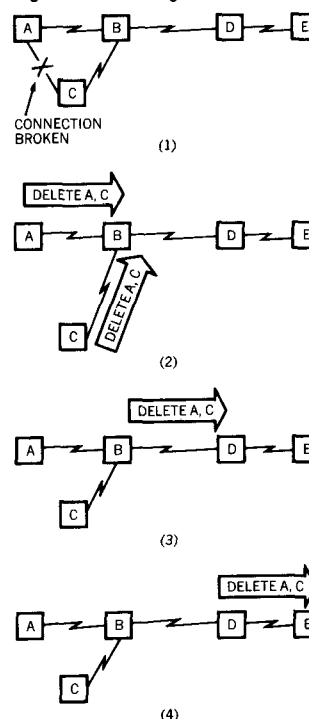
five-node network. To ensure that the path managers at all nodes have the same picture of the network, the path managers at nodes B and D pass along all new information (including information about the connections between nodes B and D) to the path managers at nodes A, C, and E. Each path manager continues this promulgation until it recognizes that information it has received is not new. Information about disconnections is promulgated in the same manner, as shown by Figure 6.

In performing its communication function, the network path manager uses the same transmission services as other JES2 components. Special interfaces into the JES2 Remote Terminal Access Method (RTAM) provide the path manager with the ability to send and receive records on a MULTI-LEAVING basis over communication lines and channel-to-channel adapters. Communication across the multiaccess spool is accomplished via the remote console services interface with the standard JES2 spooling services.

Although the function of the network path manager is not necessarily unique to NJE, allowance has been made for systems without the full path manager dynamic path determination capability to participate in the NJE network. It is through a subset of the full NJE protocols that NJI systems (VM/370, HASP, and ASP) connect to NJE. The NJI subset involves the exchange of sign-on records when two nodes become connected, but not the promulgation of connection events. The NJI systems use prebuilt routing tables (preassembled for HASP, built at system initialization time for VM/370 and ASP) to describe the job network. These tables may be modified by the NJI operators during network operation to allow for changes in the network configuration, but they are not updated automatically, and they revert to their prebuilt forms at the next system initialization. Because the NJI subset of the network path manager protocols does not support the promulgation of connection events, connections involving NJI nodes must be predefined to NJE during NJE initialization, in much the same manner as the routing tables are defined for VM/370 and ASP.

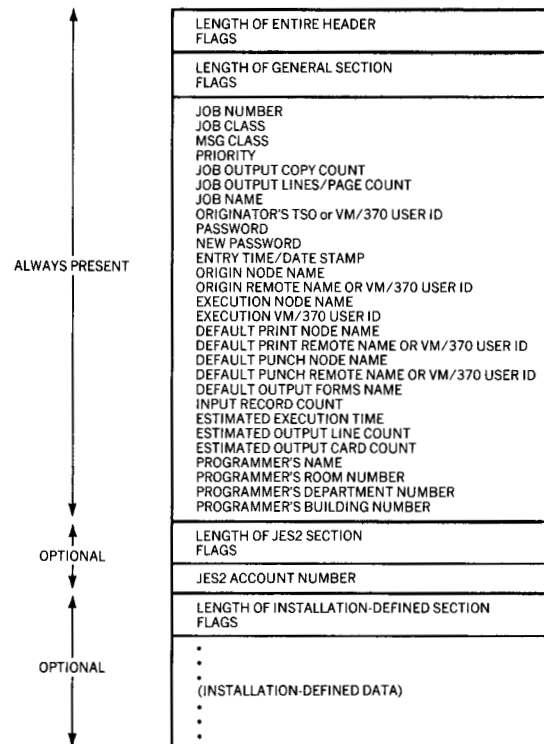
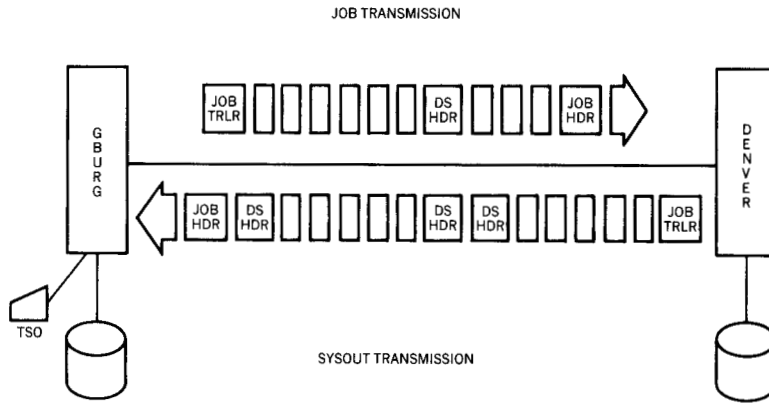
The function of the NJE job transmitter is to transmit to an NJE job receiver the JCL file and in-stream data set files that make up a job before its execution (Figure 7). If JES2 is to communicate with non-JES2 systems such as HASP or VM/370, its transmission technique cannot depend on the manner in which the input stream for a job is handled internally by JES2. For this reason, when transmitting a job, the job transmitter sends a job header record containing descriptive information about the job and then sends the JCL file with any in-stream files reinserted in their original locations between JCL statements. The data stream sent by the job transmitter is identical to the original input stream of the job, except each in-stream data set is preceded by a data set header record describing its characteristics (e.g., logical record length).

Figure 6 Breaking a connection



job
transmitter

Figure 7 Job and system output transmission



The job header (Figure 8) is not reconstructed for each transmission of the job or its output. Rather, the original job header is stored and forwarded with the job. Each node that handles the job may add information to the header, but all the original information is maintained, including user-supplied accounting information. When the job has been completely transmitted and acknowl-

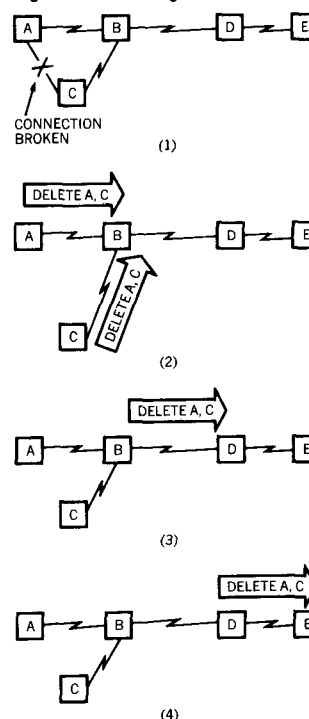
five-node network. To ensure that the path managers at all nodes have the same picture of the network, the path managers at nodes B and D pass along all new information (including information about the connections between nodes B and D) to the path managers at nodes A, C, and E. Each path manager continues this promulgation until it recognizes that information it has received is not new. Information about disconnections is promulgated in the same manner, as shown by Figure 6.

In performing its communication function, the network path manager uses the same transmission services as other JES2 components. Special interfaces into the JES2 Remote Terminal Access Method (RTAM) provide the path manager with the ability to send and receive records on a MULTI-LEAVING basis over communication lines and channel-to-channel adapters. Communication across the multiaccess spool is accomplished via the remote console services interface with the standard JES2 spooling services.

Although the function of the network path manager is not necessarily unique to NJE, allowance has been made for systems without the full path manager dynamic path determination capability to participate in the NJE network. It is through a subset of the full NJE protocols that NJI systems (VM/370, HASP, and ASP) connect to NJE. The NJI subset involves the exchange of sign-on records when two nodes become connected, but not the promulgation of connection events. The NJI systems use prebuilt routing tables (preassembled for HASP, built at system initialization time for VM/370 and ASP) to describe the job network. These tables may be modified by the NJI operators during network operation to allow for changes in the network configuration, but they are not updated automatically, and they revert to their prebuilt forms at the next system initialization. Because the NJI subset of the network path manager protocols does not support the promulgation of connection events, connections involving NJI nodes must be predefined to NJE during NJE initialization, in much the same manner as the routing tables are defined for VM/370 and ASP.

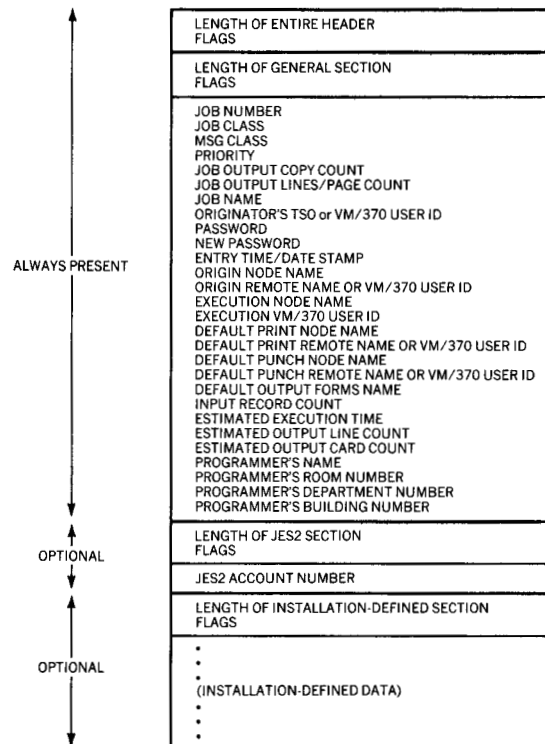
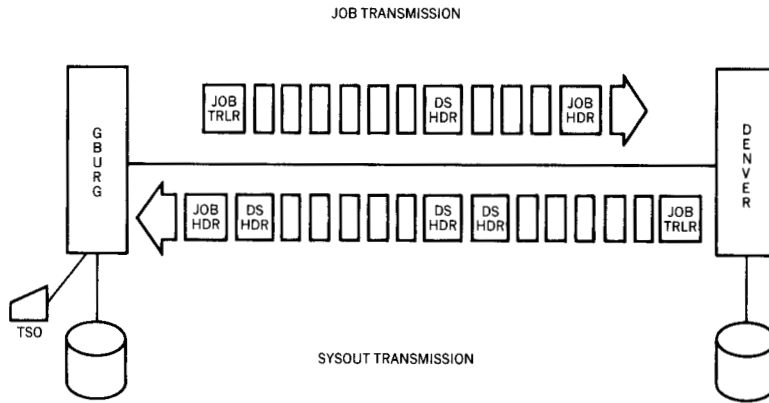
The function of the NJE job transmitter is to transmit to an NJE job receiver the JCL file and in-stream data set files that make up a job before its execution (Figure 7). If JES2 is to communicate with non-JES2 systems such as HASP or VM/370, its transmission technique cannot depend on the manner in which the input stream for a job is handled internally by JES2. For this reason, when transmitting a job, the job transmitter sends a job header record containing descriptive information about the job and then sends the JCL file with any in-stream files reinserted in their original locations between JCL statements. The data stream sent by the job transmitter is identical to the original input stream of the job, except each in-stream data set is preceded by a data set header record describing its characteristics (e.g., logical record length).

Figure 6 Breaking a connection



job
transmitter

Figure 7 Job and system output transmission



The job header (Figure 8) is not reconstructed for each transmission of the job or its output. Rather, the original job header is stored and forwarded with the job. Each node that handles the job may add information to the header, but all the original information is maintained, including user-supplied accounting information. When the job has been completely transmitted and acknowl-

edgment has been received, the job is removed from the queue of JES2 at the transmitting node, and the spool space occupied by its files is made available for reuse.

The output stream produced by the job transmitter is the input stream to the job receiver at the next node. Because the original input stream of the job is recreated by the job transmitter, the function of the job receiver is virtually the same as that of the regular JES2 input service routine. As the job is received, the JCL is placed in one file, and the in-stream data sets are placed in separate files. When all of the data for the job has been received, acknowledgment is sent to the transmitting node, and the execution node name specified for the job is examined. If the execution node is the same as that of the job receiver, the job is queued for the OS/VS2 JCL converter and eventual execution. If the job is to be executed on some other node, the job is queued for transmission; eventually it will be selected by a job transmitter for further transmission.

**job
receiver**

Once a job has completed execution, its appearance to JES2 changes markedly. Now it consists of a collection of system output files (the JCL and input files are ignored by JES2 from this point on). No longer is the job queued by JES2 as an individual and complete entity; rather, various groups of the system output files of the job are placed on different JES2 queues based on characteristics of the system output files such as destination and special forms requirements. Each group of files results in a separate work element on a JES2 output queue. The manner in which a job is queued after execution is strongly dependent on the internal workings of JES2. The function of the NJE system output transmitter is to transmit an appropriate set of the system output files of the job in a manner independent of the internal structure of JES2 (Figure 7).

**system
output
transmitter**

Each system output transmitter is supplied a list of nodes to which it is eligible to transmit, as determined by the network path manager. When selecting a job for transmission, a transmitter selects from the output queues of JES2 all those work elements representing output files for the same job that are destined for any of the nodes to which the transmitter is eligible to send a transmission. Thus, for a single job, as the first step on the paths to their ultimate destinations, all the output files that may be sent to the same directly connected node are transmitted together, even though their ultimate destinations may all be different. This procedure reduces the number of individual transmissions resulting from a single job and the number of individual jobs created at intermediate nodes that receive the system output of the job.

Having selected a set of output work elements for a job, the system output transmitter sends a job header to describe the job to

the jobs that belong to him without having to know the command languages supported by the different systems that make up the job network.

The following global commands have been defined:

- Display or locate job.
- Hold job.
- Release job from hold.
- Cancel job.
- Reroute job.

JES2 supplies its own versions of all of these commands for the operator to use when controlling local JES2 jobs. By using the global commands to control jobs at other nodes, the JES2 operator can still enter all commands using the familiar JES2 command syntax. If an operator needs to enter a command that is not covered by the list of global commands, JES2 can send any command text wanted to any other node. To do this, however, the operator must be familiar with the operating system and job entry subsystem at the other node, and must enter the command using the syntax supported by the other node.

The NJE operator controls the way in which a node communicates with the rest of the job network by using a set of JES2 commands to start and stop telecommunications lines, and to connect to and disconnect from other nodes. The job and system output transmitters and receivers associated with each communications line can be controlled in the same manner as local and remote I/O devices—any transmitter or receiver can be started, stopped, restarted, or canceled. Finally, the attributes of each node in the job network can be changed as seen from the operator's node: for example, passwords can be specified for use when connecting with other nodes, the level of command authority that another node is allowed when sending commands to the operator's node can be specified, and all jobs that originated at a specified node can be held before execution.

**programmer
control**

The programmer, or the one who submits a job for execution, can specify that the job is to be executed at another location by including an NJE route control card (/ROUTE XEQ) in the JCL for the job. The programmer is not aware of the path the job may take while being transmitted to the execution node; only the destination is coded on the control card.

Routing defaults of JES2 are usually set up so that if the job submitter does nothing more than include a route control card, the job is run on the specified node and the output is returned to the job's node of origin. If desired, the submitter can specify that an individual system output file be sent to a different destination

(anywhere in the network) by coding the destination name on the JCL statement defining the file. Collections of system output files generated by a job may be routed as a group to one or more different destinations by coding the destination names on an output control card. The job submitter may change the default print destination and default punch destination (which apply to system output files for which no specific destination is coded) by coding the destination on the proper route control cards for printing or punching.

Destinations coded on route and output control cards may specify not only a node name but also a remote work station name or VM/370 user identifier. Thus, a VM/370 user may submit a job to NJE for JES2 and have the output of the job returned to the virtual card reader on a VM/370 virtual machine.

A TSO user under OS/VS2 can submit a job to NJE for execution at any node in the job network by including a route control card within the job submitted. Unless specified differently using JCL or JES2 control cards, the output from the job will be returned to NJE at the user's node. If desired, the submitter may specify that the system output files for the job be held for access by the TSO user from his terminal. Such "held" files are transmitted by NJE to their ultimate destination where they are held until the TSO user disposes of them.

TSO support

If a TSO user requests notification about events concerning the job submitted, the terminal is sent a message whenever the job ends execution, no matter where that may be in the job network. In addition, if the job is to be run at some other node, notification is given when the local node (the one to which the TSO user is connected) transmits the job for execution.

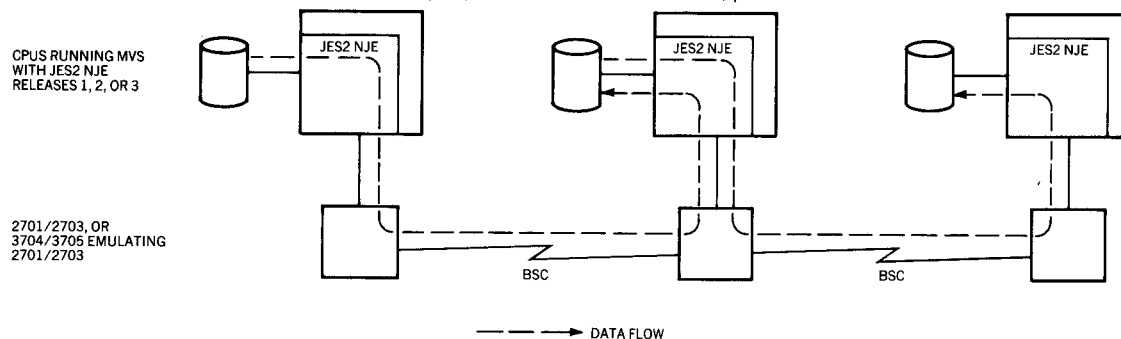
Notification is also given when any of the system output files of the job are received at their ultimate destination. The commands that the TSO user can issue have not been extended to the job network—inquiry about a job from a TSO user results in a "no jobs found" message while the job is away from the TSO user's node.

NJE writes accounting records through the OS/VS2 System Management Facility (SMF) for all processing that is performed for a job. In addition to regular accounting records for job input, execution, and output, new records are written to account for job transmission and reception and system output transmission and reception.

accounting

The accounting information is collected by the local accounting routines at each node that performs processing for a job. To facilitate charging for processing at different nodes for the same job,

Figure 9 Store-and-forward intermediate spooling required with NJE Release 1 or 2, possible with Release 3



NJE provides a mechanism for translating account numbers as a job passes from one node to another. A network account number has been defined and is carried in the NJE job header which always accompanies a job and its system output. When the job is first submitted to JES2, the local account number specified on the job card is translated to a network account number using an installation-defined conversion table. The network account number is placed in the job's NJE job header. As the job is transmitted from one node to another, the (unchanging) network account number is translated to a local account number at each node using the installation-defined conversion table of that node. This technique allows a job network to be constructed of systems that use different account number schemes without forcing each system to conform to a network-wide standard.

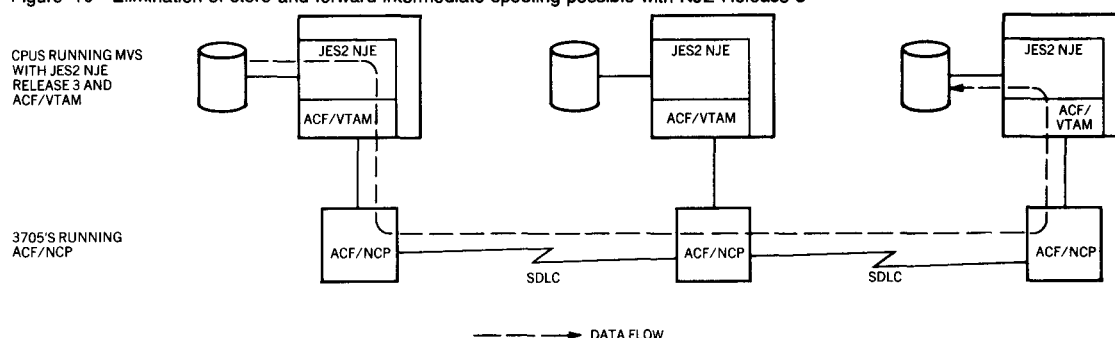
Extensions to NJE for systems network architecture

The third release of NJE for JES2 extends its communications capabilities. Support added in Release 3 makes it possible for NJE to use the Advanced Communications Functions of VTAM and NCP (Network Control Program) to communicate with NJE running on other OS/VS2 systems. To VTAM and NCP, NJE is an application using the standard application program interface to communicate with other applications (also NJE) at other locations. This support for the SNA Advanced Communications Function in NJE is in addition to the support already provided for BSC lines and channel-to-channel adapters; the older support remains for compatibility and migration purposes.

SNA communications advantages

The use of VTAM and NCP for NJE communications produces three main advantages over BSC: link sharing, full-duplex communications, and the elimination of store-and-forward intermediate spooling. These advantages are discussed below.

Figure 10 Elimination of store-and-forward intermediate spooling possible with NJE Release 3



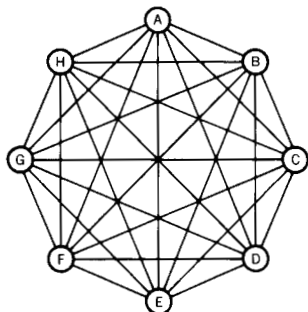
Link sharing. The MULTI-LEAVING protocol used by JES2 for communications on BSC lines and channel-to-channel adapters demands a point-to-point connection between two nodes that cannot be shared with any other application. Thus, if nodes are connected by a BSC line for NJE, no other communications may flow on that line. If communications between other applications on the two nodes are needed, then parallel lines must be installed.

With SNA communications, parallel lines are not necessary. The physical teleprocessing line between the two nodes is under the control of NCP running in the IBM 3705 Communications Controller; any application (including NJE) at one node can communicate with an application at another node by interfacing through VTAM, which communicates with NCP. Control of the physical teleprocessing line is removed from the application program, and sharing of the line is controlled by NCP.

Full-duplex communications. With SNA communications, the synchronous data link control (SDLC)¹⁰ lines between NCPs are driven in a full-duplex manner: data flows in both directions at once on the two halves of the full-duplex line. Although the MULTI-LEAVING protocol used by NJE on BSC lines allows data streams to flow in both directions, data is never actually transmitted in both directions simultaneously. Rather, the MULTI-LEAVING protocol sends a block of data in one direction and receives a block of data in return as a response. The genuine full-duplex communications provided by the control of NCP of the SDLC lines allows greater utilization of a teleprocessing line because one half of the line is not idle while the other half is transmitting.

Elimination of store-and-forward intermediate spooling. In a job network of any complexity (more than three or four nodes), it is unusual for there to be a physical teleprocessing line connecting

Figure 11 SNA NJE network with sessions established between all possible pairs of NJE nodes

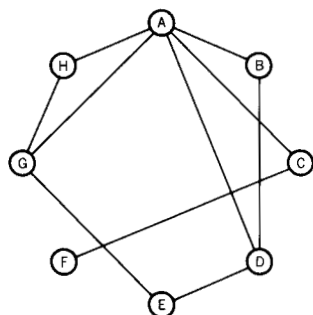


each node with all other nodes. Instead, the network is rather sparsely connected such that jobs and system output data sets pass through several intermediate nodes to reach their final destination. At each intermediate node, the job or system output data is written to the JES2 spool; when all of the data has been written out, transmission to the next node in the path can begin (Figure 9). This procedure results in three major disadvantages: an increase in the amount of space required for the JES2 spool at the intermediate nodes, increased CPU cycles required for reception and retransmission at the intermediate nodes, and a dramatic increase in queuing delay that a job experiences as it travels through more and more intermediate nodes.

With the advent of application-to-application communications through VTAM and NCP, the intermediate spooling and its inherent overhead and delay can be eliminated. An SNA session, or logical connection, can be established directly between two NJE systems, though there may be no direct teleprocessing line between them. Data transmitted by the first system flows through VTAM at its node, to the directly connected NCP, through NCPs at one or more intermediate nodes, to the NCP directly connected to the second node, through VTAM at that node, finally into the second NJE system (Figure 10). The CPUs at the intermediate NCP nodes are not involved in this connection at all—it is not necessary for the intermediate nodes to have NJE installed for this type of communication to work. To NJE, it appears that a direct connection has been established between the two NJE nodes; NJE has no knowledge of the configuration of the SNA network through which it communicates.

**job
networking
philosophy
changes**

Figure 12 SNA NJE network with infrequently used sessions omitted



NJE views the SNA session that is established between two NJE systems as a virtual "line." That is, it appears to be a communication line that runs directly from NJE on one end to NJE on the other end; the fact that it runs through VTAM and any number of intermediate NCPs does not matter. This view of the SNA session as a "line" allows NJE to keep its network path manager virtually unchanged; the establishment of an SNA session is treated the same way as is the connection of a BSC line. As a result, a job network can consist of any desired mixture of SNA sessions (using VTAM and NCP), BSC lines (driven directly by JES2), and channel-to-channel adapters (also driven directly by JES2). The ability to mix connections in this manner is mandatory for migration from BSC to SNA and for allowing older systems such as HASP to remain in a job network that is being converted to SNA.

If sessions are established between every possible pair of NJE nodes in an SNA network, it appears to NJE that a direct line exists from each node to every other node: no store-and-forward operations at all are required (Figure 11). However, in a large network not all nodes may wish to communicate with all other nodes all of

the time. Nodes that have only infrequent traffic between them need not establish a direct session (Figure 12). Because the network path manager knows of all the connections (sessions) between NJE nodes, the store-and-forward function through intermediate NJE nodes is invoked automatically if a node has a job or system output destined for a node with which it does not have an SNA session. The store-and-forward mechanism is also the way in which job entry subsystems that do not have SNA support pass data into the SNA NJE network: jobs and system output are passed to NJE over a BSC line or channel-to-channel adapter, and the NJE system then transmits them through the SNA network.

To convert to SNA NJE from BSC NJE it is not necessary to install NJE for JES2 Release 3 at all nodes in a network at the same time. Since Release 3 can communicate with all previous releases of NJE using BSC lines, nodes in the NJE network can be converted to SNA NJE one at a time. As soon as two nodes have been converted, they can start using SNA communications between themselves while maintaining BSC communications with the rest of the network.

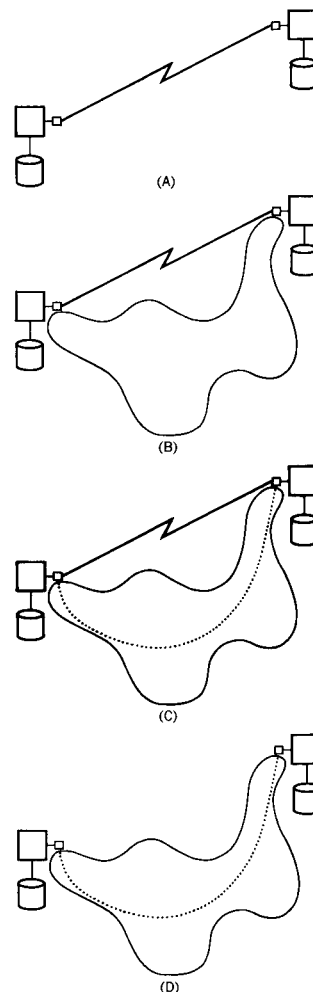
In fact, it is not necessary to remove any of the BSC lines in operation to install SNA communications. An SNA session can be established in parallel (logically) with an existing BSC line (Figures 13A through 13D). JES2 would then have two parallel links between the two nodes—one BSC and one SNA. When the performance of the SNA session has been established, the BSC line can be removed. Eventually the entire NJE network can be converted to SNA one piece at a time. When all of the JES2-to-JES2 communications have been converted to SNA, the BSC support is still available for connections to older operating systems and job entry subsystems.

Summary

The Network Job Entry Facility for JES2 provides a simple yet powerful means of connecting computer systems together. It allowed a user to move into application-to-application networking even before the availability of this type of support in SNA. It is easy to install: system programmers who are familiar with the non-networking versions of JES2 (Releases 4.0 and 4.1) will find that moving to the NJE facility is not a large step. In conjunction with NJI, NJE allows job networks to be built from computers running four different types of operating systems and job entry subsystems; this type of network can be both a migration aid and a powerful interconnection between interactive systems (such as VM/370) and batch systems (such as MVS). Finally, NJE provides a clear-cut and easy-to-use migration path from BSC networking to SNA.

conversion to SNA

Figure 13 NJE link migration: (A) initial connection—BSC NJE link, (B) first step—establish SNA/SDLC link, (C) next step—start SNA NJE session, (D) final step—eliminate BSC link



CITED REFERENCES

1. J. H. Baily, J. A. Howard, and T. J. Szczygalski, "The job entry subsystem of OS/VS1," *IBM Systems Journal* **13**, No. 3, 253-269 (1974).
2. *HASP-II (Version 3.1)*, 360D-05.1.014, IBM Corporation, Program Information Department, Hawthorne, NY 10532.
3. J. H. McFadyen, "Systems Network Architecture: An overview," *IBM Systems Journal* **15**, No. 1, 4-23 (1976).
4. *Introduction to Advanced Communications Function*, GC30-3033, IBM Corporation, Data Processing Division, White Plains, NY 10604.
5. H. R. Albrecht and K. D. Ryder, "The Virtual Telecommunications Access Method: A Systems Network Architecture perspective," *IBM Systems Journal* **15**, No. 1, 53-80 (1976).
6. W. S. Hobgood, "The role of the Network Control Program in Systems Network Architecture," *IBM Systems Journal* **15**, No. 1, 39-52 (1976).
7. *OS/VS2 JES2 Logic*, SY28-0622, IBM Corporation, Data Processing Division, White Plains, NY 10604.
8. R. P. Crabtree, "Job networking," *IBM Systems Journal* **17**, No. 3, 206-220 (1978) this issue.
9. *Network Job Interface, General Information: PRPQ P09007 (VM/370 Networking Prog 5799-ATB), PRPQ P09008 (ASP Networking Prog 5799-ATC), PRPQ P09009 (HASP Networking Prog 5799-ATC)*, GH20-1941, IBM Corporation, Data Processing Division, White Plains, NY 10604.
10. R. A. Donnan and J. R. Kersey, "Synchronous data link control: A perspective," *IBM Systems Journal* **13**, No. 2, 140-162 (1974).

Reprint Order No. G321-5072.